7

8

9

10

11

12

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

- 1 1. (Currently Amended) In a communication network, a method of TCP state 2 migration comprising the steps of:
- a) establishing a TCP/IP communication session between a client computer
 and a first server computer, said first server computer part of a plurality of server
 computers forming a web cluster containing information, said communication session
 established for the transfer of data contained within said information;
 - b) handing off said communication session to a selected server computer from said first server computer over a persistent control channel using TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer, that wherein the TCP handoff modules implement a TCP handoff protocol that works within kernel levels of an existing TCP/IP protocolthe operating systems; and
- c) migrating a first TCP state of said first server computer to said selected
 server computer, and a second TCP state of said selected server computer to said first
 server computer over said control channel.

1 2. (Currently Amended) The method as described in Claim 1, wherein said 2 step a) comprises the steps of: 3 receiving a SYN packet from said client computer at a first BTCP bottom TCP 4 (BTCP) module located at said first server computer; 5 sending said SYN packet upstream to a first TCP module located above said first 6 BTCP module in a first operating system of said first server computer; 7 receiving a first SYN/ACK packet from said first TCP module; 8 parsing said first initial-TCP state from said first SYN/ACK packet, including a 9 first initial sequence number for said first TCP module associated with said TCP/IP 10 communication session: 11 sending said SYN/ACK packet to said client computer; 12 receiving an ACK packet from said client at said first BTCP module; 13 sending said ACK packet to said first TCP module; 14 receiving a web request packet associated with said TCP/IP communication 15 session at said first BTCP module at said first server computer; storing said SYN packet, said ACK packet and said web request packet at said 16 17 first server computer.

said first BTCP module.

1	3. (Currently Amended) The method as described in Claim 2, wherein said	
2	step b) comprises the steps of:	
3	examining content of said web request packet;	
4	determining which of said plurality of server computers, a selected server	
5	eomputer, can best process said WEB-web request packet, based on said content, wherei	n
6	the server computer determined to be able to best process said web request packet is the	
7	selected server computer;	
8	sending a handoff request packet from said first BTCP module to a second BTCI	P
9	module at said selected server computer over said control channel, if said selected server	r
10	computer is not said first server computer;	
11	including said SYN packet and said ACK packet in said handoff request packet;	
12	changing a first-destination IP address of said SYN packet to a second IP address	s
13	of said selected server computer, at said second BTCP module;	
14	sending said SYN packet from said second BTCP module to said a second TCP	
15	module at said selected server computer;	
16	receiving a second SYN/ACK packet at said second BTCP module;	
17	parsing said second initial-TCP state from said second SYN/ACK packet,	
18	including a second initial sequence number, for said second TCP module, that is	
19	associated with said TCP/IP communication session;	
20	changing a second-destination IP address of said ACK packet to said second IP	
21	address, at said second BTCP module;	
22	updating said ACK packet to reflect said second TCP state of said selected serve	r
23	computer in said communication session;	
24	sending said ACK packet that is updated from said second BTCP module to said	
25	second TCP module; and	
26	sending a handoff acknowledgment message from said second BTCP module to	

1 4. (Currently Amended) The method as described in Claim 3, wherein step c)
2 comprises the steps of:

monitoring traffic associated with establishing said TCP/IP communication session in step a), at said first BTCP module, to parse a first initial TCP state of said first server computer, said first initial TCP-state associated with said TCP/IP communication session; and

migrating said first initial-TCP state to said second BTCP module over said control channel by including said first initial TCP state-sequence number in said handoff request packet, said first initial TCP state including a first sequence number; such that said second BTCP module can calculate said first TCP state for said first server computer in said TCP/IP communication session.

5. (Currently Amended) The method as described in Claim 3, wherein step c)
 comprises the steps of:

monitoring traffic associated with handing off said TCP/IP communication session at said second BTCP module, to parse a second initial TCP state of said selected server computer, said second initial TCP state associated with said TCP/IP communication session; and

migrating said second initial-TCP state of said selected server computer to said first BTCP module by including said second initial TCP state sequence number in said handoff acknowledgment packetmessage, said second initial TCP state including a second initial sequence number, such that said first BTCP module can calculate said second TCP state for said selected server computer in said TCP/IP communication session.

6.

further steps of:

1 2

3

4 5

6

7	holding said connection indication message at said first UTCP module, wherein
8	said first UTCP module and said first BTCP module provide a wrapper around said first
9	TCP module.
1	7. (Currently Amended) The method as described in Claim 6, wherein said
2	method comprises the further steps of:
3	sending a reset packet from said first BTCP module upon receiving said handoff
4	acknowledgment packet message to said first TCP module;
5	discarding said connection indication message at said first UTCP module;
6	receiving incoming data packets from said client at said first BTCP module;
7	changing said-destination addresses of said incoming data packets to said second
8	IP address;
9	updating sequence numbers and TCP checksum in said data packets to reflect said
10	second TCP state of said selected server computer; and
11	forwarding said <u>updated</u> data packets to said selected server computer.
1	8. (Currently Amended) The method as described in Claim 6, comprising the
2	further steps of:
3	sending notification from said first BTCP module to said first UTCP module to
4	release said connection indication message, if said selected server computer is said first
5	server computer;
6	sending incoming data packets, including said web request packet, from said
7	client computer, received at said first BTCP module, upstream.

(Currently Amended) The method as described in Claim 2, comprising the

intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module at

the first server computer, said connection indication message sent by said first TCP

module upon establishing said communication session; and

Appl. No.: 09/880,631
Amendment Dated: January 7, 2010
Reply to Office Action mailed October 7, 2009

	reply to office rectal manded october 1, 2005
1	9. (Currently Amended) The method as described in Claim [[1]]2,
2	comprising the further stepsteps of:
3	intercepting outgoing response packets from said selected server computer at a
4	second bottom TCP (BTCP) module located at said selected server computer;
5	changing, by the second BTCP module, source addresses of said response packets
6	to a first IP address of said first server computer;
7	updating, by the second BTCP module, sequence numbers and TCP checksum in
8	said response packets to reflect said first TCP state of said first server computer; and
9	sending said updated response packets to said client computer without passing the
10	updated response packets through the first server computer.
1	10. (Currently Amended) The method as described in Claim [[1]]9,
2	comprising the further steps of:
3	monitoring TCP/IP control traffic for said communication session at said second
4	BTCP module;
5	understanding when said communication session is closed at said second-selected
6	server computer;
7	sending a termination message to said first server computer over said control
8	channel;
9	terminating said TCP/IP communication session at said first server computer by
10	terminating a forwarding mode at said first BTCP module; and
11	freeing data resources associated with said communication session at said first
12	server computer.

4

5

6

7

8

9

10

11

12

13

14

15

18

19

20

21

22

23

24

25

26

27

28

- 11. (Currently Amended) In a communication network, a method of TCP state
 migration comprising the steps of:
 - establishing a TCP/IP communication session between a client computer and a first server computer, said first server computer part of a plurality of server computers forming a web cluster containing information, said communication session established for the transfer of data contained within said information;
 - b) monitoring traffic associated with establishing said TCP/IP communication session to understand a first initial TCP state of said first server computer associated with said TCP/IP communication session, at a first bottom-TCP (BTCP) module at said first server computer, wherein said first BTCP module is a dynamically loadable kernel module (DLKM) loaded in said first server computer without modifying a first operating system of said first server computer:
 - receiving a web request associated with said TCP/IP communication session at said first BTCP module at said first server computer;
 - d) examining content of said web request;
- e) determining which of said plurality of server computers, a selected server
 computer, can best process said web request, based on said content;
 - f) handing off said communication session to said selected server computer from said first server computer over a persistent control channel, if said selected server computer is not said first server computer;
 - g) monitoring traffic associated with handing off said TCP/IP communication session to understand a second initial TCP state of said selected server computer associated with said TCP/IP communication session, at a second BTCP module at said selected server computer, wherein said second BTCP module is a DLKM loaded in said selected server computer without modifying a second operating system of said selected server computer:
 - migrating said first initial TCP state to said selected server computer over said control channel, such that said second BTCP module can calculate a first TCP state for said first server computer in said TCP/IP communication session;

34

35

36

37

38

39

- i) sending a second initial TCP state of said selected server computer to said
 first BTCP module, such that said first BTCP module can calculate a second TCP state
 for said selected server computer in said TCP/IP communication session;
 - j) forwarding data packets received at said first BTCP module from said client to said selected server computer, by changing said data packets to reflect said second TCP state and a second IP address of said selected server computer;
 - sending response packets from said selected server computer directly to said client computer by changing said response packets to reflect said first TCP state and a first IP address of said first server computer; and
 - terminating said TCP/IP communication session at said first server computer when said TCP/IP communication session is closed.
- 1 12. (Currently Amended) The method as described in Claim 11, wherein said 2 stepsteps a) and b) comprisecomprises the steps of:
- 3 receiving a SYN packet from said client computer at said first BTCP module;
- sending said SYN packet upstream to a first TCP module located above said first
 BTCP module in [[a]]said first operating system of said first server computer;
- 6 receiving a first SYN/ACK packet from said first TCP module;
- 7 parsing, by the first BTCP module, said first initial TCP state from said first
- 8 SYN/ACK packet, including a first initial sequence number for said first TCP module
- 9 associated with[[,]] said TCP/IP communication session;
- sending said SYN/ACK packet to said client computer;
- 11 receiving, by the first BTCP module, an ACK packet from said client computer at
 12 said first BTCP module:
- 13 sending said ACK packet to said first TCP module;
- 14 storing said SYN packet, said ACK packet and said web request at said first
- 15 server computer.

	17
1	13. (Currently Amended) The method as described in Claim [[11]]12, wherein
2	said step e)steps f) and g) comprise comprises the steps of:
3	sending a handoff request packet from said first BTCP module to said second
4	BTCP module over said control channel;
5	including said SYN packet and said ACK packet in said handoff request packet;
6	changing a first-destination IP address of said SYN packet to a second IP address
7	of said selected server computer, at said second BTCP module;
8	sending, by said second BTCP module, said SYN packet to said a second TCP
9	module_at said selected server computer;
10	receiving a second SYN/ACK packet at said second BTCP module;
11	parsing said second initial TCP state from said second SYN/ACK packet,
12	including a second initial sequence number, for said second TCP module, that is
13	associated with said TCP/IP communication session;
14	changing a second-destination IP address of said ACK packet to said second IP
15	address, at said second BTCP module;
16	updating said ACK packet to reflect said second TCP state of said selected server
17	computer in said communication session;
18	sending said ACK packet that is updated from said second BTCP module to said
19	second TCP module; and
20	sending a handoff acknowledgment message $\underline{\text{from said second BTCP module}}$ to
21	said first BTCP module.
1	14. (Currently Amended) The method as described in Claim 13, wherein said

(Currently Amended) The method as described in Claim 13, wherein said ACK packet includes said first initial TCP state of said first server computer-as provided for in step f).

Appl. No.: 09/880,631	
Amendment Dated: January 7, 2010	
Reply to Office Action mailed October 7, 200	9

	Reply to Office Action mailed October 7, 2009
1	15. (Currently Amended) The method as described in Claim 13, wherein said
2	handoff acknowledgment message includes said second initial TCP state of said second
3	server computer, including a second initial sequence number, for said second TCP
4	module, that is associated with said TCP/IP communication session as provided for in
5	step i).
1	16. (Currently Amended) The method as described in Claim 13, comprising
2	the further steps of:
3	intercepting a connection indication message sent from said first TCP module to
4	an application layer above said first TCP module at a first upper-TCP (UTCP) module \underline{at}
5	said first server computer, said connection indication message sent by said first TCP
6	module upon establishing said communication session; and
7	holding said connection indication message at said first UTCP module, wherein
8	said first UTCP module and said first BTCP module provide a wrapper around said first
9	TCP module.
1	17. (Currently Amended) The method as described in Claim 16, wherein step
2	h) comprises comprising the further steps of:
3	sending a reset packet from said first BTCP module upon receiving said handoff
4	acknowledgment packet message to said first TCP module;
5	discarding said connection indication message at said first UTCP module;
6	receiving incoming data packets from said client at said first BTCP module;
7	changing said-destination addresses of said incoming data packets to said second
8	IP address;
9	updating sequence numbers and TCP checksum in said data packets to reflect said

forwarding said updated data packets to said selected server computer.

second TCP state of said selected server computer; and

Appl. No.: 09/880,631
Amendment Dated: January 7, 2010
Reply to Office Action mailed October 7, 2009

	Reply to Office Action mailed October 7, 2009
1	18. (Currently Amended) The method as described in Claim 11, wherein step
2	k) comprises the steps of:
3	intercepting outgoing response packets from said selected server computer at said
4	second BTCP module;
5	changing, by said second BTCP module, source addresses of said response
6	packets to said first IP address;
7	updating, by said second BTCP module, sequence numbers and TCP checksum in
8	said response packets to reflect said first TCP state of said first server computer; and
9	sending said updated response packets to said client computer without passing the
10	updated response packets through said first server computer.
1	19. (Currently Amended) The method as described in Claim 11, wherein step
2	1) comprises the steps of:
3	monitoring TCP/IP control traffic for said communication session at said second
4	BTCP module;
5	understanding when said communication session is closed at said second-selected
6	server computer;
7	sending a termination message to said first server computer over said control
8	channel;
9	terminating a forwarding mode at said first BTCP module; and
10	freeing data resources associated with said communication session at said first
11	server computer.

- 1 20. (Currently Amended) The method as described in Claim 16, comprising 2 the further steps of:
- sending notification from said first BTCP module to said first UTCP module to
 release said connection indication message, if said selected server computer is said first
 server computer; and
- server computer; and
- sending incoming data packets, including said web request, from said client
 computer, received at said first BTCP module, upstream.

- 21. (Currently Amended) The method as described in Claim 11, wherein each of said plurality of server computers is constructed similarly including BTCP modules located downstream from respective TCP modules in the corresponding server computers, and UTCP modules located upstream from the corresponding TCP modules, wherein each pair of the BTCP modules and UTCP modules provides a wrapper around a
- 1 22. (Cancelled)

corresponding TCP module.

- 1 23. (Currently Amended) The method as described in Claim [[22]]21, wherein
 2 said control channel allows for communication between all UTCP modules of the
 3 corresponding server computers.
- 1 24. (Original) The method as described in Claim 11, wherein said plurality of
 2 server computers is coupled together over a wide area network in said communication
 3 network.
- 1 25. (Original) The method as described in Claim 11, wherein said information 2 is partitioned/partially replicated throughout each of said plurality of server computers.

Amendment Dated: January 7, 2010
Reply to Office Action mailed October 7, 2009

26. (Currently Amended) A <u>first</u> server computer comprising:

an operating system:

a TCP module in the operating system;

an-a first upper TCP (UTCP) module located above [[a]]the TCP module in
[[an]]the operating system of said <u>first</u> server computer;

a <u>first</u> bottom TCP (BTCP) module located below said TCP module, <u>wherein</u> said

first UTCP, TCP, and first BTCP modules implement implementing a method of handing
 off a communication session between the first server computer and a second node,
 wherein the first BTCP and first UTCP modules are dynamically loadable kernel modules

(DLKMs) that are dynamically loadable and unloadable in said first server computer

without modifying the operating systema first node and second node in a cluster network
 that works within the kernel level of an existing TCP/IP protocol, by migrating TCP

13 states associated with said first and second nodes.

Appl. No.: 09/880.631

1

2

3

4

5

	7.33
1	27. (Currently Amended) The <u>first</u> server computer as described in Claim 26,
2	wherein said-method comprises the steps of:
3	a) establishing a TCP/IP communication session is established between a
4	client computer and said server computer, said first node, said first server computer part
5	of a plurality of server computers $\underline{including\ said\ second\ node\ }$ forming $\underline{said\ \underline{a}\ }\underline{c}luster\underline{that}$
6	$\underline{contains} \ \underline{network} \ \underline{containing} \underline{.} \underline{information}, \underline{said} \ \underline{communication} \ \underline{session} \ \underline{established} \ \underline{for} \ \underline{the}$
7	transfer of data contained within said information;
8	b) receiving wherein said first BTCP module is configured to:
9	receive a web request associated with said TCP/IP communication session
10	at a first BTCP module at said server computer;
11	 e) examining-examine content of said web request;
12	 determining determine which of said plurality of server computers,
13	a selected server computer, can best process said web request, based on said content,
14	wherein the server computer determined to be able to best process said web request is the
15	second node;
16	e) handing hand off said communication session to said selected
17	server computer second node from said first server computer over a persistent control
18	channel, if said $\underline{\text{selected server computer}}\underline{\text{second node}}$ is not said $\underline{\text{first}}$ server computer,
19	wherein the handing off causes migration of a first TCP state of said first server computer
20	to the second node, and migration of a second TCP state of said second node to the server
21	computer; and
22	f) migrating a first TCP state of said server computer to said selected server
23	${\color{red} \textbf{computer}, \textbf{and sending a second TCP state of said selected server \textbf{computer to said server}}$
24	computer over said control channel.

Appl. No.: 09/880,631
Amendment Dated: January 7, 2010
Reply to Office Action mailed October 7, 2009

1	28. (Currently Amended) The first server computer as described in Claim 27,
2	wherein said first BTCP module is configured to furtherstep a) of said method comprises
3	the steps of:
4	receiving-receive a SYN packet from said client computerat said BTCP module;
5	sending-send said SYN packet upstream to said TCP module;
6	receiving-receive a first SYN/ACK packet from said TCP module;
7	parsing parse a first initial TCP state from said first SYN/ACK packet, including a
8	first initial sequence number for said TCP module associated with said TCP/IP
9	communication session;
10	sending send said SYN/ACK packet to said client computer;
11	receiving-receive an ACK packet from said client computer at said BTCP module;
12	sending send said ACK packet to said TCP module;
13	storing said SYN, ACK at said server computer.
1	29. (Currently Amended) The <u>first</u> server computer as described in Claim 28,
2	wherein said <u>first BTCP module is configured to further</u> method comprises the steps of:
3	sending send a handoff request packet from said BTCP module to a second BTCP
4	module over said control channel, said second BTCP module located below a second
5	TCP module in a second operating system at said second nodeselected server computer;
6	including include said SYN packet and said ACK packet in said handoff request
7	packet;
8	receiving-receive a handoff acknowledgment message at said BTCP module from
9	said second BTCP module.
1	30. (Cancelled)

Appl. No.: 09/880,631
Amendment Dated: January 7, 2010
Reply to Office Action mailed October 7, 2009

1	31. (Currently Amended) The <u>first</u> server computer as described in Claim 29,
2	wherein said <u>first UTCP module is configured to method comprises the further steps of</u> :
3	intercepting intercept a connection indication message sent from said first TCP
4	module to an application layer above said first-TCP module at a first upper-TCP (UTCP)
5	module, said connection indication message sent by said first TCP module upon
6	establishing said communication session; and
7	holding hold said connection indication message at said first UTCP module.
1	32. (Currently Amended) The eomputer first server system as described in
2	Claim 31, wherein said first BTCP module is configured to further method comprises the
3	further steps of:
4	sending send a reset packet from said first BTCP module upon receiving said
5	handoff acknowledgment message packet to said first TCP module;
6	discarding said connection indication message at said first UTCP module;
7	receiving receive incoming data packets from said client computer at said first
8	BTCP module;
9	changing said change destination addresses of said incoming data packets to said
10	a second IP address of said second node;
11	updating update sequence numbers and TCP checksum in said data packets to
12	reflect said second TCP state of said second nodeselected server computer; and
13	forwarding forward said updated data packets to said second nodeselected server
14	computer.
1	33. (Currently Amended) The <u>first</u> server computer as described in Claim 31,
2	wherein said first BTCP module is configured to furthersaid method comprising the
3	further steps of:
4	sending send notification from said BTCP module to said first UTCP module to
5	release said connection indication message, if said second nodeselected server computer
6 7	is said <u>first</u> server computer;
/	sending-send incoming data packets, including said web request, from said client

1	34. (Currently Amended) The <u>first</u> server computer as described in Claim 26,
2	$\underline{wherein\ said\ first\ BTCP\ module\ is\ configured\ to} \underline{said\ method\ comprising\ the\ further\ steps}$
3	of:
4	receiving receive a handoff request from a first second BTCP module located at a
5	$\underline{first}\underline{second}\underline{server}computer\underline{within}saidcluster\allowbreaknetwork\underline{over}apersistentcontrolchannel,$
6	said <u>first-second</u> server computer having established a <u>second</u> communication session
7	with a \underline{second} client computer, \underline{said} communication session established for the transfer of
8	data contained within said server computer, said handoff request including a SYN packet
9	and an ACK packet, said SYN_packet and ACK packet used for establishing said second
10	$communication\ session\ between\ said\ \underline{second}\ client\ \underline{computer}\ and\ said\ \underline{first}\underline{second}\ server$
11	computer, said ACK packet including a first initial TCP state $\underline{including}$ a first initial TCP
12	sequence number of said first-second server computer in said communication session,
13	including a first initial TCP sequence number;
14	changing change a first destination IP address of said SYN packet to a second-first
15	IP address of said first server computer, at said BTCP module;
16	sending-send said SYN packet to said TCP module;
17	receiving-receive a SYN/ACK packet-at-said second BTCP module;
18	parsing-parse a second initial TCP state from second-said_SYN/ACK packet,
19	including a second initial sequence number, for said TCP module, said second initial TCP
20	state associated with a second TCP state for said server computer in said TCP/IP
21	communication session;
22	ehanging-change a second-destination IP address of said ACK packet to said
23	second-first IP address, at said BTCP module;
24	updating update said ACK packet to reflect said second TCP state-of said selected
25	server computer in said communication session;
26	sending-send said ACK packet that is updated to said TCP module; and
27	sending_send_a handoff acknowledgment message to said first_second_BTCP
28	module over said control channel.

1 35.-37. (Cancelled)

- 1 38. (New) The method as described in Claim 1, wherein the TCP handoff
 2 modules are dynamically loadable kernel modules, the method further comprising:
 3 dynamically loading the TCP handoff modules in the corresponding first server
 4 computer and selected server computer without modifying the operating systems of the
 5 respective first server computer and selected server computer.
- 1 39. (New) The method as described in claim 6, wherein the first BTCP module and first UTCP module are dynamically loadable kernel modules.